Extract by User's Manual VES-MATIC 30 / 30Plus

DESCRIPTION OF THE REFERENCES:

8) Main switch (ON/OFF)
9) Fuses and voltage change
10) Power cord socket
11) RS-232 connector for the external computer
12) Connector for external bar code reader
13) RS485 connector
14) Forced ventilation fan

Rear View

7.2. CONNECTION A HOST COMPUTER TO VES 20/30 INSTRUMENTATION

Check that the connection cables are 3 wire connectors (Tx-Rx-Gnd) and are connected as follows:

VES CONNECTOR	HOST-COMPUTER CONNECTOR	
Type DB9	Type DB9	Type DB25
2	2	3
3	3	2
5	5	7

7.3. SPECIFICATIONS FOR THE ASYNCHRONOUS SERIAL COMMUNICATION PROTOCOL

The new generation of Ves-Matic line supports two type of serial communication protocol: unidirectional protocol, for compatibility with the old Ves-Matic line (OLD TYPE) and bidirectional protocol (NEW TYPE)
The old type protocol is selected in the default configuration (contact the technical service to change the configuration).

7.3.1. SPECIFICATION OF THE UNIDIRECTIONAL COMMUNICATION PROTOCOL –"OLD TYPE "

When the instrument is switched on, the following string of 5 characters is send to the Host Computer : [CR]Com>

During performance of the analysis, the instrument transmits the hard-copy of the data sent to the printer with the following specifications.

An echo is made of all the lines of the heading sent to the printer before the results are printed, terminating with the NULL and CR characters; lastly, a string of SPACE characters terminating with NULL and CR characters.
Below are the lines containing the results, starting with CR character.

The records are therefore of 2 types:
Type 1 STRING-NULL-CR
Type 2 CR-STRING

where:

STRING :	Sequence of 24 printable ASCII characters.
CR:	Carriage Return (ASCII 0x0D)
NULL:	Null (ASCII 0x00)
SPACE:	(ASCII 0x20)

Example of display on the Host Computer:

```
        VES MATIC 30                (**)
**********************              (**)
CYCLE---------------------- 1       (**)
SELECT--------------------F1        (**)
TEMPERATURE----- OFF                (**)
Q.C.----------------------ON        (**)
DATE(mm/dd/yy) 10/22/97             (**)
TIME(hh:mm:ss)- 10:21:33            (**)

1  =  SAMPLE ABSENT                 (*)
2  =  ------------------------ 1    (*)
..  =  ------------------------ --   (*)
..  =  ------------------------ --   (*)
30= SAMPLE LOW                      (*)
```

(*) Lines of print of which a type 2 echo is made
(**) Lines of print of which a type 1 echo is made

Example of program receive data from HOST-COMPUTER

Below is an example of software (written in basic) apt to receive data from serial port during analysis

EXAMPLE N°1

```
100 OPEN "com1:9600,n,8,1,cs,ds,cd" FOR RANDOM AS #1     'Open RS232 com1
110 OPEN "spool.txt" FOR OUTPUT AS #2                    'Open capture file
120 IF NOT EOF(1) THEN                                   'Start of loop
130 a$ = INPUT$(1,1)                                     'Extracts characters
140 IF a$ = CHR(3) THEN GOSUB 200                        'If character CR
elaborates
150 PRINT #2,a$;                                         'Update file
160 END IF                                               'End of loop
170 IF INKEY$ = CHR$(27) THEN 190                        'Exit if ESC pressed
180 GOTO 120                                             'Continue loop
200 REM                                                  'Routine of elaboration
        ...         ...
        ...         ...
        ...         ...
        ...         ...
500 RETURN
```

EXAMPLE N°2

```
CLS
OPEN "COM2:9600,n,8,1,cs,ds,cd" FOR RANDOM AS #1      'Open RS232 COM2
OPEN "spool20.txt" FOR OUTPUT AS #2                   'Open capture file
DO                                                    'Start of loop
IF INKEY$ = CHR$(27) THEN EXIT DO                     'Exit if ESC pressed
IF NOT EOF(1) THEN
  a$ = INPUT$(1, 1)                                   'Read character
  PRINT a$;                                           'View character
  PRINT #2, a$;                                       'Record character
END IF
LOOP                                                  ‘Continue loop
CLOSE
```

The following figure shows the print of the file, obtained with the previous programs, containing all the characters sent by the Ves-Matic20 during analysis:
(this example is also valid for the Ves-matic 30 but with a difference: the total number of samples, instead 20 or 15, will be 30)

```
00000000   20 20 20 20  20 20 56 45  53 20 4D 41  54 49 43 20        VES MATIC
00000010   32 30 20 20  20 20 20 20  00 0D 2A 2A  2A 2A 2A 2A  20      .******
00000020   2A 2A 2A 2A  2A 2A 2A 2A  2A 2A 2A 2A  2A 2A 2A 2A  **************
00000030   2A 2A 00 0D  43 59 43 4C  45 20 5F 5F  5F 5F 5F 5F  **.CYCLE ______
00000040   5F 5F 5F 5F  5F 5F 5F 5F  5F 20 20 32  00 0D 53 45  _________  2.SE
00000050   4C 45 43 54  20 5F 5F 5F  5F 5F 5F 5F  5F 5F 5F 5F  LECT ___________
00000060   5F 5F 5F 20  46 31 00 0D  54 45 4D 50  45 52 41 54  ___ F1.TEMPERAT
00000070   55 52 45 20  5F 5F 5F 5F  5F 5F 5F 5F  20 4F 46 46  URE ________OFF
00000080   00 0D 51 2E  43 2E 20 5F  5F 5F 5F 5F  5F 5F 5F 5F  .Q.C. __________
00000090   5F 5F 5F 5F  5F 5F 20 4F  46 46 00 0D  44 41 54 45  ______ OFF.DATE
000000A0   20 28 4D 4D  2F 44 44 2F  59 59 29 20  30 38 2F 32  (MM/DD/YY) 08/2
000000B0   37 2F 39 37  00 0D 54 49  4D 45 20 28  48 48 3A 4D  7/97.TIME (HH:M
000000C0   4D 3A 53 53  29 20 30 39  3A 35 30 3A  35 38 00 0D  M:SS) 09:50:58.
000000D0   20 20 20 20  20 20 20 20  20 20 20 20  20 20 20 20
000000E0   20 20 20 20  20 20 20 20  00 0D 0D 20  20 31 20 3D           .  1 =
000000F0   20 2E 2E 2E  2E 2E 2E 2E  2E 2E 2E 2E  2E 2E 20 20   ..............
00000100   20 31 20 0D  20 20 32 20  3D 20 53 41  4D 50 4C 45  1    2 = SAMPLE
00000110   20 41 42 53  45 4E 54 20  20 20 20 20  0D 20 20 33  ABSENT        3
00000120   20 3D 20 2E  2E 2E 2E 2E  2E 2E 2E 2E  2E 2E 2E 2E  = ..............
00000130   20 20 20 31  20 0D 20 20  34 20 3D 20  53 41 4D 50  1     4  = SAMP
00000140   4C 45 20 41  42 53 45 4E  54 20 20 20  20 20 0D 20  LE ABSENT
00000150   20 35 20 3D  20 2E 2E 2E  2E 2E 2E 2E  2E 2E 2E 2E  5 = ...........
00000160   2E 2E 20 20  20 31 20 0D  20 20 36 20  3D 20 53 41  .. 1     6 = SA
00000170   4D 50 4C 45  20 41 42 53  45 4E 54 20  20 20 20 20  MPLE ABSENT
00000180   0D 20 20 37  20 3D 20 53  41 4D 50 4C  45 20 48 49    7 = SAMPLE HI
00000190   47 48 20 20  20 20 20 20  20 0D 20 20  38 20 3D 20  GH          8 =
000001A0   53 41 4D 50  4C 45 20 41  42 53 45 4E  54 20 20 20  SAMPLE ABSENT
000001B0   20 20 0D 20  20 39 20 3D  20 53 41 4D  50 4C 45 20       9 = SAMPLE
000001C0   41 42 53 45  4E 54 20 20  20 20 20 0D  20 31 30 20  ABSENT       10
000001D0   3D 20 53 41  4D 50 4C 45  20 41 42 53  45 4E 54 20  = SAMPLE ABSENT
000001E0   20 20 20 20  0D 20 31 31  20 3D 20 2E  2E 2E 2E 2E       11 = .....
000001F0   2E 2E 2E 2E  2E 2E 2E 2E  20 20 20 31  20 0D 20 31  ........  1   1
00000200   32 20 3D 20  53 41 4D 50  4C 45 20 41  42 53 45 4E  2 = SAMPLE ABSEN
00000210   54 20 20 20  20 20 0D 20  31 33 20 3D  20 2E 2E 2E  T       13 = ...
00000220   2E 2E 2E 2E  2E 2E 2E 2E  2E 2E 20 20  20 31 20 0D  ..........   1
00000230   20 31 34 20  3D 20 53 41  4D 50 4C 45  20 41 42 53   14 = SAMPLE ABS
00000240   45 4E 54 20  20 20 20 20  0D 20 31 35  20 3D 20 53  ENT      15 = S
00000250   41 4D 50 4C  45 20 41 42  53 45 4E 54  20 20 20 20  AMPLE ABSENT
00000260   20 0D 20 31  36 20 3D 20  53 41 4D 50  4C 45 20 41    16 = SAMPLE A
00000270   42 53 45 4E  54 20 20 20  20 20 0D 20  31 37 20 3D  BSENT     17 =
00000280   20 2E 2E 2E  2E 2E 2E 2E  2E 2E 2E 2E  2E 2E 20 20   ..............
00000290   20 31 20 0D  20 31 38 20  3D 20 53 41  4D 50 4C 45  1   18 = SAMPLE
000002A0   20 41 42 53  45 4E 54 20  20 20 20 20  0D 20 31 39  ABSENT       19
000002B0   20 3D 20 53  41 4D 50 4C  45 20 41 42  53 45 4E 54  = SAMPLE ABSENT
000002C0   20 20 20 20  20 0D 20 32  30 20 3D 20  53 41 4D 50        20 = SAMP
000002D0   4C 45 20 41  42 53 45 4E  54 20 20 20  20 20        LE ABSENT
```

Note: If the QC is introduced, the QUALITY CONTROL string is not preceded by the CR character.

7.3.2. SPECIFICATION OF THE BIDIRECTIONAL COMMUNICATION PROTOCOL – "NEW TYPE "

Representation of the bytes transmitted on the serial line:

STX	BLK	BLK	LEN	LEN	ADD	ADD	COM	COM	D1	...	Dn	ETX	CHK	CHK

Each box represents 1 byte transmitted on the serial port. The pair of bytes will instead be packed in the memory and return to the value of 1 byte.

STX : Character ASCII '>' 0x3E

BLK : Block Number, always = '00' excepted multi-block transfers (see command 03)

LEN : Number of characters, from D1 to Dn (included)

ADD : Device ID (00 and FF = Reserved) Available range 01-7F

COM : Command ID

If COM=COM+0x80, the end of the block is given by ETX and CHK is not controlled.

D1.Dn : Command related data

ETX : Character ASCII 'CR' 0x0D

CHK : CheckSum (XOR from STX to Dn included)

The reply to the protocol will have to be part of the receiving device:

ACK / NAK	ADD	ADD	ETX

ACK : Character ASCII 0x06

If the command block has been correctly interpreted according to the protocol specifications

NAK : Character ASCII 0x15

If the command block is not compliant with protocol specifications

ADD : Device ID

If data transaction requires the use of more blocks, these will be numbered in progressive order starting from 0. Every transmission of more blocks should end with a block with a length equivalent to zero.

7.4. VES20/30 SERIAL COMMANDS AND PARAMETER FORMAT

Each command is interpreted and receives as reply 'ACK'=[0x06]+ID+CR
If a syntax error occurs or an incorrect command is issued, the reply will be 'NACK'=[0x15]+ID+CR

0x01: Request of version

Example:

From Host

>00000181+CR+00 *Asks unit with ID '01' to run command '81' (that is command '01' without checksum control)*

From ESR

>00190101VES MATIC 20New Rel. 1.00+CR+10

0x02: Request list of stored tests

The returned data field is formed by the number of tests stored in ASCII hex format (2 bytes) followed by 15 byte separated by ';' which specify the type, date and time of the test.

Example:

From Host

>00000182+CR+00 *Asks unit with ID '01' to run command '82' (that is command '02' without checksum control)*

From ESR

>0031010203F1 28/04 15:41;F2 28/04 15:49;F2K 28/04 16:13+CR+15

0x03: Request test transmission

Example:

From Host

>0002018303+CR+00 *Asks unit with ID '01' to run command '83' (that is command '02' without checksum control) with parameter '03' indicating the number of tests that should be received:* **from '01' to '03' for Vesmatic30 New; from '01 to '04' for Vesmatic 20 New**. **To obtain the last analysis the parameter must be '00'.**

From ESR

The reply can be formed by more blocks, depending on the quantity of information present in the test. **NOTE: after receiving each block the host must send an ACK message.**

The unit transmits a 20 byte header with information regarding

- *Type of test 1 byte (0x01=F1, 0x02=F2, 0x03=F1K, 0x04=F2K)*
- *ESR settings during the reception 1byte (X X BO BE BI FP FD FT where FT= temperature correction flag 1 is ON, FD=Displayed results flag, FP=Printed results flag, BI=Internal bar code selection flag, BE=External bar code selection flag, BO= Barcode deselection flag)*
- *Number of samples 1 byte*
- *Cycle 1byte*
- *Temperature during test 1 byte*
- *Test date 10 byte hex-ascii*
- *Test time 5 byte hex-ascii*

Followed by a number of 40 byte fields equivalent to the "Number of samples" with the following format:

(Note: each 128 bytes ESR wait ACK and after sends a new Block!)

- *Position number 1 byte*
- *Status flag 1 byte*
- *Barcode 13 byte Hex-ascii*
- *ESR results 24 byte*
- *Katz index 1 byte*

Flag Status can acquire the following values:

- *0x81 Abnormal cuvette*
- *0x82 High cuvette*
- *0x84 Low cuvette*
- *0x88 Empty cuvette*
- *0x00 Ordinary cuvette*

If the test carried out is of type F1, the ESR results will be equivalent to the first byte of the 24. The Katz index shall not be taken into account.

If the test carried out is of type F2, the ESR results will be equivalent to the first two bytes of the 24. The Katz index will also be taken into account

If the test carried out is of type F1K, the ESR results will be equivalent to the first 12 bytes of the 24. The Katz index shall not be taken into account

If the test carried out is of type F2K, the ESR results will be within the first 24 bytes. The Katz index will also be taken into account.

If the test carried out is of type F1- FAST, the ESR results will be equivalent to the first byte of the 24. The Katz index shall not be taken into account.

If the test carried out is of type F2-FAST, the ESR results will be equivalent to the first two bytes of the 24. The Katz index will also be taken into account

0x04: Command to request the unit status

The unit status is constituted by 2 ASCII hex integers represented by means of two sets, each with 4 bytes.

The *first four* bytes contain the following information:

Bit value:

Bit	Code
0	Test type
1	Test type
2	Test type
3	Reset in progress
4	Check -Device expired
5	Open cover
6	Sample reading in progress
7	Mixing in progress
8	Centrifugation in progress
9	Test aborted
10	Error condition
11	Last analysis ready to be sent
12	- -
13	- -
14	- -
15	- -

The three *Type of test* bits 0,1 and 2 represent:

- 0x01 F1 Normal
- 0x02 F2 Normal
- 0x03 F1 Kinetic
- 0x04 F2 Kinetic
- 0x05 F1 fast
- 0x06 F2 fast

The second group of 4 bytes represents the time in seconds needed to complete the test in progress.

Example:

From Host

Send command >00000184+CR+00 00 Ask the unit with ID '01' to run command '0x84' (that is command '0x04' without checksum control)

From ESR

>00080104008105CD+CR+4D

The status is codified as 0x0081, that is an ordinary F1 test in progress and a mixing stage in progress. The seconds needed to complete the test are 0x05CD, that is 1485

seconds.

0x05: Reading the Setting register

The unit returns the value of the setting register in 2 ASCII hex bytes.

The register bit have a Boolean value (1=ON)

Bit	Code
0	Temperature correction
1	Displayed results
2	Printed results
3	Internal barcode
4	External barcode
5	Disabled barcode
6	- -
7	- -

Example:

From Host

Send command >00000185+CR+00 00 Ask the unit with ID '01' to run command '0x85' (that is command '0x05' without check sum control)

From ESR

>0002010525+CR+3F

Consequently: temperature correction is ON, Printed results is ON and the Barcode is disabled.

0x06: Writing in the Setting register

This command enables to set the Setting register.

Example:

From Host

>0002018682+CR+00 Ask the unit with ID '01' to run command '0x86' with data field 0x82 (that is command '0x06' without checksum control). This sets: Temperature correction OFF, Displayed results ON, Printed results OFF, Internal barcode ON.

From ESR

*Replies **'ACK'** for positive results; otherwise it replies **'NACK'***

0x07: Start test

The command enables to start and select the type of test.

The type is codified as follows:

Type	Code
0x01	F1 Normal
0x02	F2 Normal
0x03	F1 Kinetic
0x04	F2 Kinetic
0x05	F1 Fast
0x06	F2 Fast

Example:

From Host

Sends command >0002018703+CR+00

Ask the unit with ID '01' to run command '0x87' with data field 0x03 (that is command '0x07' without checksum control). This start the F1 kinetic test.

From ESR

*Replies **'ACK'** for positive results; otherwise it replies **'NACK'***

0x08: Block analysis

The command enables to block the test in progress.

Example:

From Host

Send command >00000188+CR+00

Ask the unit with ID '01' to run command '0x88' (that is command '0x08' without checksum control).

From ESR

*Replies **'ACK'** for positive results; otherwise it replies **'NACK'***

0x09: Read Barcode

It reads the barcode stored in the requested position.

Positions range from 0 to 19 for VES20.

Example:

From Host

Send command >000201890D+CR+00

Ask the unit with ID '01' to run command '0x89' (that is command '0x09' without check sum control) with data field 0x0D, that is ask the barcode to be sent to position 13.

From ESR

>000D0109 . . . . . +6C

The units sends the 13 character code related to the requested barcode.

0x0A: Write barcode

This command enables the write the barcode in the desired position during the test cycle.

Positions range from 0 to 19 for VES20.

Example:

From Host

Send command >000F018A07ABCDEFGHILMNO+CR+00

Ask unit with ID '01' to run command '0x8A' (that is command '0x0A' without checksum control) with data field 07ABCDEFGHILMNO. This enables to write the bar code "ABCDEFGHILMNO" in position 0x07.

From ESR

'ACK'** if the reply is positive; otherwise **'NACK'

0x0B: Read Date and Time

This command enables to read the internal timer of the unit. The reply is formatted with 6 ASCII hex characters (12 byte) that refer to: hour, minutes, seconds, date, month and year.

Example:

From Host

Send command >0000018B+CR+00

Ask unit with ID '01' to run command '0x8B' (that is command '0x0B' without checksum control).

From ESR

>000C010B0B14040C0C00+CR+4D

The data field provides the following data: 11:20:04 hours with date 12/12/00

0x0C: Set date and time

This command enables to set the internal timer of the unit. The command data field is formatted with 6 ASCII hex characters (12 byte) that refer to: hour, minutes, seconds, date, month and year.

Example:

From Host

Send command >000C018C0C00000F0601+CR+00

Ask unit with ID '01' to run command '0x8C' (that is command '0x0C' without checksum control) with data field 0C00001F0601. This enables to set the timer at 12:00:00 hours and with a 15/06/01 date .

From ESR

'ACK'** if the reply is positive; otherwise **'NACK'

0x0D: Read Check-Device

The command returns the value of the formatted Check-Device as ASCII hex integer (4 byte).

Example:

From Host

Send command >0000018D+CR+00

Ask unit with ID '01' to run command '0x8D' (that is command '0x0D' without checksum control).

From ESR

>0004010D0F99+CR+39

The Check-Device value is 0x0F99, that is 3993